Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 13 (2018) pp. 125-136

A Blockchain-based Approach for Access Control in eHealth Scenarios

João Pedro Dias[1,2], **Ângelo Martins**[2] **and Hugo Sereno Ferreira**[1,2]

[1]Department of Informatics Engineering
Faculty of Engineering, University of Porto Porto, Portugal
{*jpmdias,hugosf*}*@fe.up.pt*

[2]INESC TEC, Porto, Portugal
angelo.martins@inesctec.pt

Abstract: **Access control is a crucial part of a system's security, restricting what actions users can perform on resources. Therefore, access control is a core component when dealing with eHealth data and resources, discriminating which is available for a certain party. We consider that current systems that attempt to assure the share of policies between facilities are prone to system's and network's faults and do not assure the integrity of policies life-cycle. By approaching this problem with a blockchain where the operations are stored as transactions, we can ensure that the different facilities have knowledge about all the parts that can act over the eHealth resources while maintaining integrity, auditability, and authenticity.**

Keywords: eHealth, Access Control, Blockchain, Distributed Ledger Technology, Security

I. Introduction

Healthcare is experiencing an explosion of data partially due to the widespread of health data collection systems such as wearables (e.g. fitness trackers) [1], health tracking applications (e.g. diet tracking) [2] and ambient assisted living systems such as CAALYX [3]. By now, it is estimated that medical data will grow at a rate of 48% per year, reaching 2.3 zettabytes by the year of 2020 [4, 5].

Lots of new smart objects are empowering the creation of cyber-physical smart pervasive systems with application in multiple domains, including healthcare [6, 7]. These smart objects, which fall under the umbrella of the Internet-of-Things (IoT), that foresees the advance towards new smart and inter-connected systems by the means of ubiquitous computing [8].

The explosion of data being collected and, *a posteriori*, analyzed by different entities, leads to the debut of data security and privacy issues [9]. By one hand, these issues are taken into account because such smart devices may be connected to the Internet at some point for accessing its collected data anytime and anywhere [8]. The data being collected from those devices may be part of the Personal Health Records (PHR) and this is typically owned by the patients and may be or not, shared with third-entities [10, 11]. On the other hand, Electronic Medical Records (EMR) and Electronic Health Records (EHR) store individual information that is required by the healthcare professionals and may be shared among different institutions and facilities [10, 11]

Hence, there is the need to control the accesses to this data resources by third-entities. Access control is concerned about determining the allowed activities of certain users, mediating every attempt by a user to access a resource in the system [12]. Dealing with the user access control to health data, personal or medical, stored by different parties, which may be required to be accessed by third-parties with different goals (e.g. insurance companies *versus* doctors), is not an easy task. This is especially problematic when we are still moving towards a unified and interoperable electronic health (eHealth) systems [13].

In this paper, we suggest an approach to the problem of access control in large scale and distributed systems, as it is observed in eHealth scenarios where different entities and users should be able to access data with different permission levels and granularities. The `Data Keepers` (e.g. hospitals, governments) should be able to manage the accesses to their data by the means of adding, changing or revoking permissions. Such a system should be also capable of defining fine-grained permissions both, at the user level and, at the resource level.

The system must also be fault tolerant, which means that it must not be dependable on a centralized architecture. Upon these considerations: consistency, integrity, and authenticity of the operations among nodes should be assured. The system must be also immutable providing an accurate audit trail.

In a previous paper, "A Blockchain-based Scheme for Access Control in eHealth Scenarios" [14], the foundation of this approach were discussed, including transactions and blocks schemes. In this paper, we push this work further. The paper is structured as follows: firstly, it is given an overview of the related work in the scope of permission management in eHealth systems, focusing also blockchain approaches for access control. Afterward, it is given a description of the purposed solution architecture. Then we address some core details of the *proof-of-concept* implementation. Finally, some final remarks are presented, summing up the contributions as well as pointing out further developments.

II. Background & State of the Art

A. Blockchain

In our solution, we take into consideration a Distributed Ledger Technology (DLT), specifically, blockchain. A distributed ledger (also known as shared ledger) consists of a consensus of replicated, shared and synchronized digital data distributed along a set of nodes, working as a distributed database, generally geographically dispersed [15]. It is important to note that, despite all blockchains being distributed ledgers, not all distributed ledgers are blockchains.

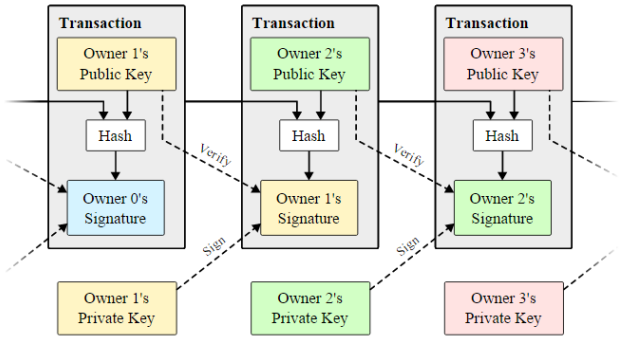

Figure. 1: Digital sign of transactions as proposed by Satoshi Nakamoto [16].

The blockchain is a specific type of distributed ledger conceptualized by Satoshi Nakamoto and used as a foundation of the digital currency Bitcoin [16]. Data in a blockchain should be tamper-proof, specifically accomplished by the use of cryptography, by the means of digital signatures and digital fingerprints (hashing), as shown in Figure 1 [15, 16]. Also, consensus must be assured among peers considering scenarios where some of the peers are providing erroneous data, by partially or completed computer/network failures or, even, by malicious intent when some party tries to subvert the ledger [15, 16].

A blockchain consists of a chain of blocks that contains information about transactions. Each one of these transactions is digitally signed by the entity emitting them. Transactions are combined into a *block*, that is committed to the chain, establishing the blockchain. Each block contains the hash of the previous block (as shown in Figure 2), being this propagated along the chain until the first block, created when the blockchain was firstly created, designed *genesis* block [15, 16].

We can then consider that a blockchain works as a state transaction system (state machine), where there is a state that corresponds to the snapshot of the chain (the result of all transaction until now) and, after adding a new block of transactions to the chain, we got a new snapshot that corresponds to a new state of the system, as result of the new transactions [18].

In order to validate a block, it is necessary a *proof-of-work*. This mechanism is used in order to get a consensus in the peer-to-peer network [16]. In Bitcoin an HashCash *proof-of-work* is used, being the work effort called *mining*. The *mining* consists of finding a *nonce* (by the means of *brute-force*) that satisfies the condition of generating a digest with the required number of leading zeroes. This *proof-of-work* guarantees consensus in a network following the principle that

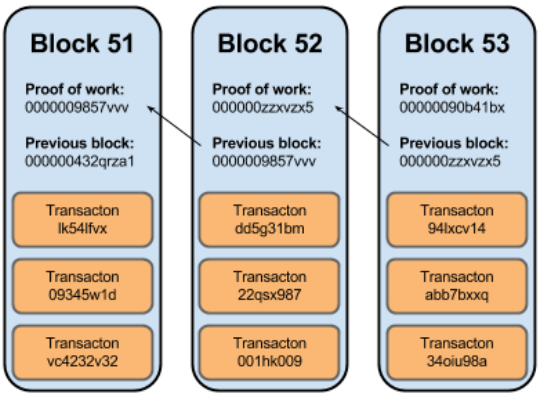

Figure. 2: Example subset of a blockchain [17].

the nodes will always accept the longest available chain [16]. This also implies that older blocks – those further back in the blockchain – are more secure than newer ones.

There are several alternatives to *proof-of-work* [19]. In the *proof-of-stake*, as it is being considered to be used in Ethereum [18], the creator of the next block to be pushed in the chain is chosen in a deterministic way based on the wealth of the node [15]. Another one, as used in the Sawtooth Lake [19], uses a *Proof of Elapsed Time* (PoET), which is a lottery-based consensus protocol that takes advantage of the trusted execution environments provided by Intel's Software Guard Extensions.

Notwithstanding the common use of blockchain for trade currencies, like Bitcoin, there exists an array of other applications for the technology. This is possible because, as blockchain is used to store *coin* transactions but it can be used to store any other domain transactions. Furthermore, it can be used as a general-purpose database distributed system, therefore making it useful in a large variety of situations [20].

Lastly, blockchains can be considered of three main kinds, as stated by Buterin [21], namely: public, fully-private and consortium. Public blockchains (e.g. Bitcoin, Ethereum), is a type of blockchain in which anyone can read, send transactions to and expect to see them included if they are valid, and, further, anyone in the world can participate in the consensus process. Fully-private blockchains consist of blockchains where write permissions are kept centralized to one organization (even if spread among facilities), existing a closed group of known participants (e.g. a supply chain) [20]. Finally, consortium blockchains, are partly private in such a way that the consensus process is controlled by a pre-selected number of nodes. In this type of blockchain, the right to query the blockchain may be public or restricted to the participants (e.g. governmental institutions and partners).

B. Access Control

The problem of access control has already been covered in the literature. We can observe different ways of controlling and managing accesses in different situations in our everyday technological systems. Issues with access control are recurrent [22], including problems in the definition of permission rules, typically known as *policies*, alongside with the problems related to inconsistency, especially in eHealth systems [23, 24].

One of the more common approaches is the use of Access Control Lists (ACL), commonly used in modern operating systems. ACL consists of a list associated with an object that specifies all the subjects that can access it, along with the access level (or rights) [12]. Other systems use Access Control Matrix, in which, each row represents a subject, each column an object and each entry is the set of access rights for that subject to that object [12].
Specifically, in healthcare, Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC) have been applied [25]. RBAC defines the users access right basing itself on his/her roles and the role-specific privileges associated with them. The ABAC system extends the RBAC role-based features to attributes, such as properties of the resource, entities, and the environment [26]. Policies in ABAC can be expressed resorting to the eXtensible Access Control Markup Language (XACML), defined by the OASIS consortium [27]. The XACML standard also includes a reference architecture for designing and implement access control systems, defining the system components and usage flow, that can be used in multiple application domains.
Another used approach to access control is the Entity-Based Access Control (EBAC) [28], which allows the definition of more expressive access control policies. This is accomplished by supporting both, the comparison of attribute values, as well as traversing relationships along arbitrary entities. Moreover, Bogaerts et al. presents *Auctoritas* as an authorization system that specifies a practical policy language and evaluation engine for the EBAC system [28].

C. Blockchain Applied to Access Control

Blockchain has been acclaimed in the literature as a panacea for the problems of controlling eHealth data, from access control to privacy [29, 30, 31].
Several approaches to resolve the access control issue based on blockchain appeared, including in eHealth scenarios. Maesa et al. [32] propose a blockchain-based access control, implementing ABAC on top of the blockchain technology, following the XACML reference architecture. This approach validates itself through a reference implementation on top of Bitcoin. However, this solution does not encompass the particularities of using such in eHealth context, namely, the possibility of having different authorities and/or entities as resource owners.
In the application of blockchain for eHealth, Yue et al. [33] proposes the use of a *Healthcare Data Gateway* (HGD) to enable the patient to own, control and share their data while maintaining data privacy. This solution also encompasses that all the patient's eHealth record is stored in a blockchain. Although the novelty of such approach, it implies a disruptive change on the already-existent systems of storing and retrieving eHealth data, what would require a considerable effort to implement which may call into question its current applicability. Also, there are cases where it is needed to access data without the explicit agreement from the patient itself (e.g. due to the patient inability to allow the access or by some governmental requirements) and this solution does not provide the ability to do such (e.g. some family member allow the data access). Also, considering the growth of eHealth data, storing this data on the blockchain itself will result in a rapid growth on its size, exceeding publicly available hard drive capacity, requiring special hardware to full nodes and, further, could lead to the centralization of the blockchain [34].

III. Illustrative Scenario

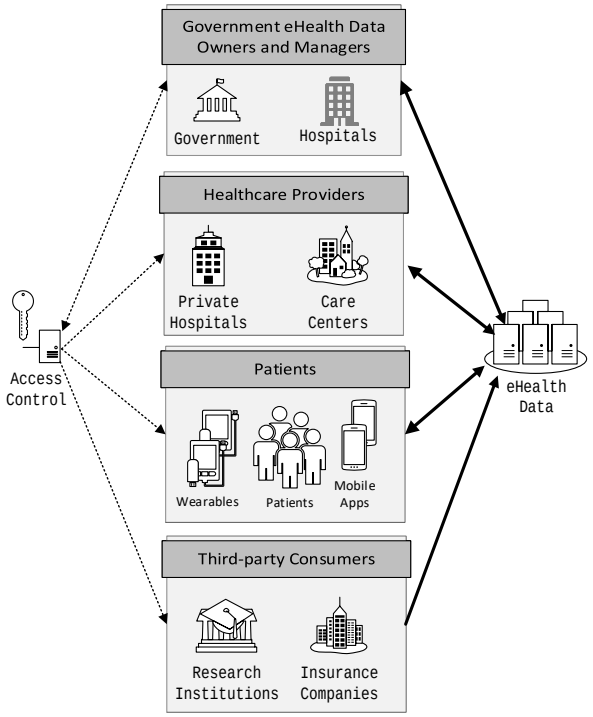

Figure. 3: An example of an eHealth system, its different participants, and typical access operations. The dashed arrows on the left show the different parties check their access rights to eHealth data, being the government the only entity that can either add, revoke or alter the access policies. The black arrows on the right of the figure show the different parties accessing eHealth data, in accordance with their respective access level.

There exists an increment of the eHealth data being produced by different sources, coming from more traditional origins like medical exams or medical reports, but, with the advent of the Internet-of-Things, *things* like wearables (e.g. fitness trackers) and ambient assisted living systems, even more, data is being produced and consumed by the individual and/or by 3rd-parties. However, it is hard to keep track of the localization of such data as well as when and who is consuming that data (Fig. 3).
As such, there must exist an access control system transversely to the eHealth domain that allows one to keep data ownership, managing easily the access policies in place. Such a system must be clear to the user, as it can easily authorize, deny or revoke permissions on-the-fly (e.g. by the means of a mobile application and push notifications).
An illustrative scenario for studying how an access control system is required to work and how it would impact the e-Health workflow was realized. A draft of such a scene is given in Figure 4. Here we can define the two major actors

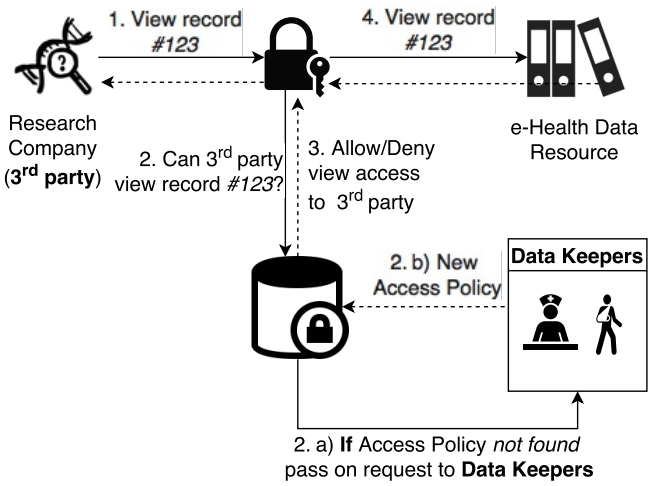

Figure. 4: An example of access to eHealth data, including its different participants. In detail, a third-party entity can request access to a given document, and, in order to do so, the access must be allowed by a request handler system. Such handler should then verify the request by checking an access control policies repository and, further, it must grant or deny the access. If some previously unknown request appears, `Data Keepers` must issue a new access control policy.

of such scenario, **Data Keepers** and **Third-Parties**.

Data Keepers is the set of individuals or entities that have ownership over a certain data-entry. They can be but are not limited to, the data creators, the institution responsible, or the data subject(s). The possibility of defining different sets of keepers is a requirement since, depending on the eHealth record or source, different restrictions may apply. Namely, when dealing with PHR data the only keeper is the individual or the legal responsible [11]. However, EMR data is controlled by a set of clinicians and staff within one health care organization, and, further, EHR data can even be shared between more than one health organization [11]. **Third-parties** are any organization, entity or individual that have interest in accessing the data (e.g. insurance companies or research institutes). Generally, these consumers have time-limited access to a small portion of data of a data-set or to a specific individual.

In detail, we can sum up the interaction of the different parties and the system as it follows (using Figure 4 as reference):

1. There is a `3rd Party` who have interest in viewing a specific eHealth record (in Figure 4 as example we use the document id *#123*). This party asks to an intermediate if it can access the record mentioned;

2. The intermediate checks the access against an Access Policy repository where it matches with already existent policies. If there is no Access Policy within a particular third-entity and a document, an extra two steps occur:

   (a) A notification with the details of the access request is sent to the set of document keepers;

   (b) The access keepers must allow or deny the access request. If there is more than one keeper on the document, a consensus must be reached. This consensus depends on the proprieties of the document and it can require the approval of one keeper, majority or all.

3. After checking the request against the Access Policies repository, the intermediate grants or denies access to the document by the 3rd-party;

4. If access is allowed, the `3rd Party` can now view the specific eHealth record (in Figure 4 the document id *#123*). If not, the request is denied and the entity cannot access the document in any way.

Taking into account the interaction flow we can say that the scenario shares the three main components of common access control systems (specified in XACML standard), namely, a Policy Decision Point (PDP), a Policy Administration Point (PAP) and a Policy Enforcement Point (PEP) [27]. Here, the PEP is the intermediate negotiator that intercepts the users' requests and enforces the PDP's decision. Further, the PDP is responsible for evaluating the requests against an Access Policies repository. However, in our case, there is not a traditional PAP, because there is no central authority managing the policies, being each one of the document keepers managers of their own documents, working as a distributed PAP system.

In this paper, we leverage the use of blockchain as a repository for Access Policies, and, furthermore, as a way of enabling the existence of a distributed PAP component.

The government eHealth data owners and managers are, in the majority of cases and for the purposes of this case study, the entities with authority to grant, revoke or manager access control to health data. For example, a hospital (government facility) should be able to grant access to patient data about a car crash to an insurance company during a certain period of time. Nonetheless, this authorization should be shared among all the government facilities and easily verifiable.

For the purposes of assuring that all the government-based facilities have enough information to allow or deny an access request at any resource, alongside with assuring fault-tolerance at the network or machine level, we propose a system based upon peer-to-peer, namely blockchain. Such an approach allows us to share all the access control policies over the network of nodes, corresponding, typically, to geographically distributed facilities (e.g. hospitals).

This is even more critical because nowadays eHealth data is dispersed among facilities, depending, for example, on their core interest/business (e.g. hospitals can have different data on their systems depending on what medical specialties they have available). For example, a patient can request write access to one government facility in order to store his AAL data, alongside with requesting other systems or facility permissions to read data about his diagnostics.

Our approach consists of using Blockchain technology as a way to accomplish a more reliable and user-empowered solution for access control management in an eHealth environment. Such an approach allows us to define fine-grained access control while maintaining the consensus in a distributed system, authenticity, immutability and auditability.

A *proof-of-concept* of the approach hereby described and detailed was implemented in order to verify its feasibility.

In our solution, we use an approach similar to the Access Control Matrix, which allows the establishment of a corre-

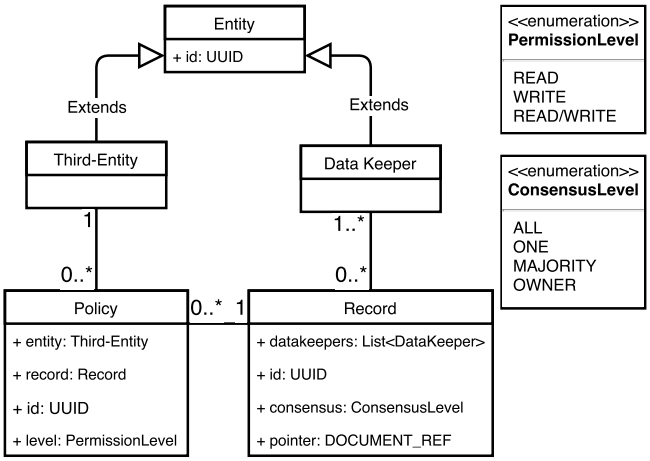

Figure. 5: UML diagram specifying the classes of the system and their relationships.

spondence between a subject, an object and a set of rights. However, this information is not stored as is, due to the inherent proprieties of the use of blockchain. As a transaction-based state machine, we store transactions corresponding to a pre-defined set of the state machine on the Access Policies repository.

A. Access Control Model

Upfront for defining our model, it is needed to define all the entities and relationships enrolled in it. Such a model is given in Figure 5, and five classes can be identified in our approach. *Entities* that can be `3rd Party`'s or `Data Keepers`. Further, we have `Policies` and `Records`. Each `Policy` refers to a relation between one and only one `3rd Party` and an eHealth `Record`, with the respective level of access, `PermissionLevel`. In turn, each `Record` can have one or more `Data Keepers` that have partial or total ownership over it.
The information related to the model is preserved by the means of storing transactions (since this model is compatible with the use of blockchain). The transactions contain information about 3 different state machines, that have dependencies between them, is always related to the class model defined, namely:

- Access Policy State Machine, as represented in Figure 6 is the state machine related to the main logic of creating Access Policies.
- Record Life-cycle State Machine, as represented in Figure 7, details the operations over an eHealth record. The record life-cycle begins with its creation, `CREATE`, then it can suffer diverse updates, `UPDATE`, until it is removed, `REMOVE`.
- Individual Authorization State Machine, as represented in Figure 8, describes the life-cycle for each user access grants over a given `REQUIRE`, which lead to a number of instantiations equal to the number of `Data Keepers` required. Each individual instantiation evokes a `REQUIRE_ACTION`, then the `Data Keeper` can allow (`AUTH_GRANT`) or deny (`AUTH_DENY`) the access. Eventually, before reaching the final state, the `Data Keeper` can revoke (`AUTH_REVOKE`) a previously granted authorization.

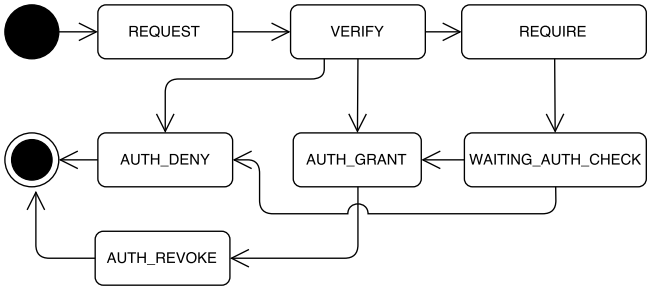

Figure. 6: State machine diagram representing the life-cycle of an access request by a `3rd Party`. The composite state `AUTH_CHECK` represents the individual authorization requests needed by the `Data Keepers` of the record being queried.

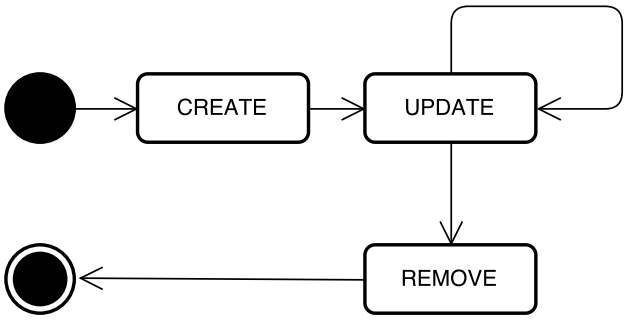

Figure. 7: State machine diagram representing the life-cycle of an eHealth record since its creation prior to becoming inaccessible.

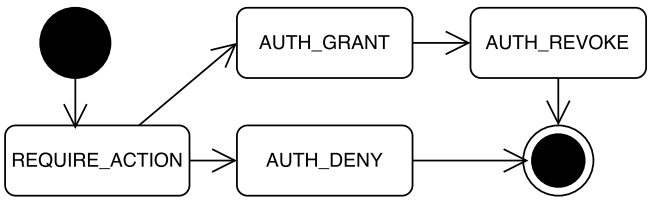

Figure. 8: State machine diagram representing the individual authorization life-cycle.

As stated, the main logic of attributing Access Policies for `3rd Party` access an eHealth record is controlled by the *Access Policy State Machine*, detailed in Figure 6. This state machine jumps the `init` state when an access request, `REQUEST` from a `3rd Party` enters the PEP. Then, the `REQUEST` is verified, `VERIFY`, against the already existent information on the blockchain (by the means of a snapshot operation). If, and only if, the information about this particular access is present in the snapshot, the request can be granted, `AUTH_GRANT` or denied `AUTH_DENY`.
Additionally, if there is no information about an access request, the Access Policy must be required, `REQUIRE`, by the means of checking the number of permissions required

to form the texttttData Keepers in order to get a consensus (specific to the record), using for that the *Individual Authorization State Machine*. This means that there is no central authority authorizing requests from 3rd Parties, and it is required that some set of Data Keepers allow the access. While this process is running, the state machine enters into a waiting state, WAITING_AUTH_CHECK.
At last, there can be a point in the future when it is needed to revoke a previously granted access, AUTH_REVOKE. The final state is, by this, reached by the existence of an AUTH_REVOKE or an AUTH_DENY.

B. Block Model

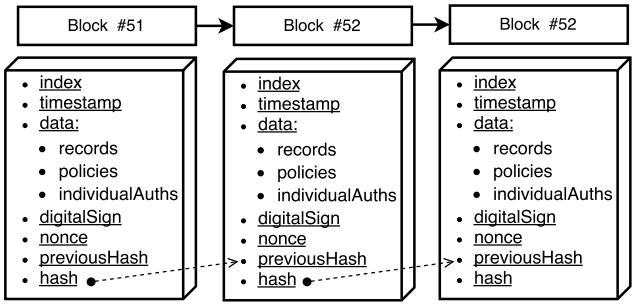

Figure. 9: An excerpt of the blockchain (consisting of the example blocks #51, #52 and #53) detailing each block content.

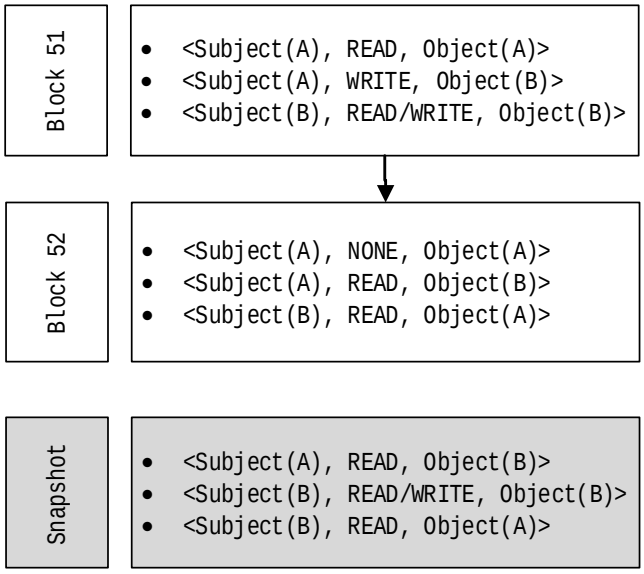

Figure. 10: Example of two system blocks, transactions data and resulting snapshot.

As our Access Control follows the model presented in Subsection III-A, based upon transactions, we leverage the use of blockchain as a repository for these transactions. This way we never have a current state of the permissions written in the repository, but, notwithstanding, we can take a snapshot (Fig. 10) to the blockchain at any moment and, as result, check the Access Policies in place.
The basic structure of a block in our chain is given in Figure 9, being each field detailed as follows:

- index: Corresponds to the index of the current block in the blockchain.
- timestamp: Timestamp corresponding to when the block was generated.
- previousHash: Hash of the previous block in the chain.
- digitalSign: Digital signature of the current block data.
- data: Content of the block. Corresponds to a set of transactions describing the access control policies, records information and individual authorizations.
- nonce: Value that is set so that the hash of the block will contain a run of leading zeros. This value is calculated iteratively until the required number of zero bits in the hash is found. This requires time and resources, making it so that the correct *nonce* value constitutes *proof-of-work*.
- hash: A SHA256 hash corresponding to the block data. This hash must have a leading *a priori* defined sequence being this leading sequence what defines the effort of the *proof-of-work*. In Bitcoin, this leading sequence corresponds to a certain number of zeros at the beginning of the hash.

Additionally, focusing on our approach, the data field should be detailed, as it is the field that serves as transaction information storage. This data field includes in it three sub-fields, namely:

- records: Transaction information relative to transactions of the state machine presented in Figure 7, about creation, update or deletion of eHealth records of any kind.
- policies: Transaction information relative to transactions of the state machine presented in Figure 6, about creation and revocation of access policies.
- individualAuths: Transactions about individual authorization by each one of a Record Data Keeper in relation to each Policy.

The use of hashes allows us to maintain integrity along the immutable chain of transactions without a central authority, since any change in the data would result in a different hash, invalidating the next blocks in the blockchain. Additionally, as a result, we can also achieve accountability and auditability. Authenticity is assured by the assign of a key-pair to each entity with access to the blockchain, identifying who write each block in the chain.

C. Architectural Design

At the system architectural level, our approach uses blockchain, being distributed by default, working as a peer-to-peer network connecting the different nodes, corresponding to the diverse facilities or organizations that can store, create or/and change eHealth data. Nodes in the network synchronize between them by following a set of rules:

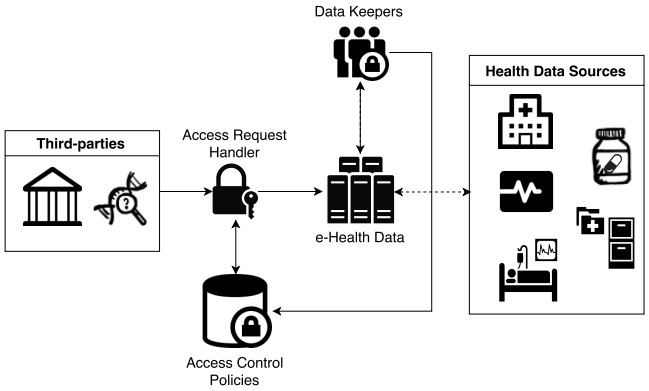

Figure. 11: Overview of the system architecture and interactions. The eHealth data access requests treated seamlessly by the handler. Additionally, the resources (eHealth data) are, typically, associated with one facility or repository, but all the available resources are known by all nodes (facilities).

- When a new block is generated by a node, this block is broadcasted to the network;
- When a node connects to a new peer in the network it queries for the latest block;
- If a node finds a block that has a higher index than the last known block, it either adds the block to its current blockchain (in the case of the difference is equal to one node) or queries another node to get the full blockchain.

The system implements a consortium blockchain[35], which means that the blockchain is partly private, in the way that instead of allowing any person to participate in the process of transaction verification or, allowing only one entity to have full control, a few selected nodes are predetermined, providing the same benefits associated with private blockchain. The `Data Keepers` and `3rd Party` can then interact with the system by the means of using public available Application Programming Interfaces (API's) or applications designed to do such.

Furthermore, despite the use of XACML standard for access control systems and blockchain as storage, there are some architectural choices due to the eHealth domain restrictions and more complex use cases. An overview can be observed in Figure 11.

Focusing on the *eHealth Data*, as of today, this data is not aggregated in one storage, being spread by multiple institutions and organizations. As such, every time that any `3rd Party` requests access to a specific record there is the need of locating this information, and then, proceed to check if the request is already approved or if there is the need to create a new access policy. So, as an improvement, information about the creation of new records must be kept and spread along all the organizations and facilities in such way that a request to a resource can be handled by any member of the private blockchain.

Additionally, aiming attention to the `Data Keepers`, it was noticed that there is a set of situations where the ownership of eHealth data records is not explicit to only one entity but shared among more than one entity or individual, as is the case of EMR *versus* PHR. Taking this into account, we set up a mechanism of consensus when creating new access policies. Each eHealth record has a level of agreement that must be achieve before allowing a `3rd Party` the access to a `Resource`, being this level associated with the `Resource` itself (Figure 5). Then, there is a number of executions of the individual authorization state machine (Figure 8) corresponding to the number of that `Record Data Keepers`. Reaching the minimum number of individual authorization (that can be either `AUTH_GRANT` or `AUTH_DENY`), the access request state machine (Figure 6) will create an access policy accordantly with the consensus reached (that can be, once more, either `AUTH_GRANT` or `AUTH_DENY`).

From the functional architecture viewpoint, we can sum up the system interaction as stated in the sequence diagram of Figure 12. This diagram describes the process of a `3rd Party` requesting access to an eHealth `Record`, owned partially or totally by one or more `Data Keepers`. Further, the diagram describes both the case of checking against an already existent access policy of the `3rd Party` and the `Record` or the process of creating a new `Policy` by checking the necessary number of `Data Keepers`.

D. Security Threat Analysis

In our approach, there are a number of security questions that must be taken into consideration since we have to consider that mistakes can be made by, for example, careless operators. Alongside with threats coming from human mistakes, there is the need for considering also faults introduced by intentional system manipulation coming from individuals with malicious intent. Note that we assume that system's operators do not have physical access to the machines where the Access Control system is deployed, communicating only by an existent GUI [1] or CLI [2].

Why existing Access Control system storage do not suffice? Currently deployed systems use a centralized or almost-centralized (*quasi*-decentralized) solution to store and access control policies, independently on how these policies are described (e.g. ACL, RBAC or EBAC). Such systems are vulnerable to network or machine failures since in case of failure all the facilities are affected by the impossibility of validating access control policy rules over their resources. The use of a DLT as a purely-distributed database for storing the policies allows the correct functioning of all non-falling facilities and automatically synchronizes when the normal function of the system is restored. Additionally, even if a node is compromised, the threat is confined to the node specifically, not compromising the whole network.

Why do Access Control systems have difficulties ensuring the validity of policies *life-cycle*? Access Control systems typical record the addition, revoking or change in policy by the means of logging. Such an approach is vulnerable to modifications by a malicious third-party because there is no way of assuring the integrity of these logs. The use of a blockchain as a way of storing policies as transactions, assures us that older policies (and operations over them) have

[1] GUI: Graphical User Interface

[2] CLI: Command-Line Interface or Command Language Interpreter.

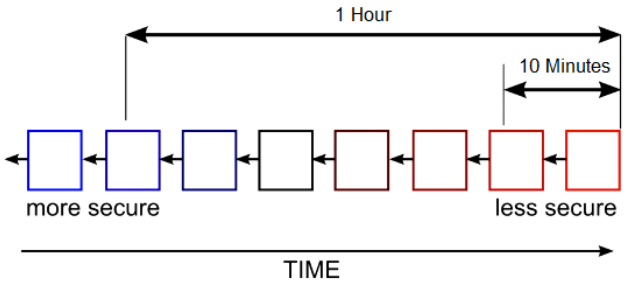

Figure. 13: Blockchain blocks security along time [17].

integrity and are frequently recomputed from a set of trusted commands. Additionally, we can ensure that, as time passes (and the blockchain grows), the blocks are more secure as you go further back in the chain.

What does it take to a third-entity with malicious intent to add a rogue policy? As the access to resources present in the eHealth system must be validated by the means of a policy in the Access Control system, successful violation by a malicious party is unconditionally dependent on the injection of rogue permission into the system. Such attempt can be mainly made by two attack vectors: (i) by attempting to mine a new block in the blockchain, though our approach makes that unlikely due to the usage of a consortium blockchain, where only allowed entities can add blocks; hence, there is the need for such malicious entity to take over more than 51% of the entire blockchain network to accomplish such task; and (ii) by compromising an Access Control system machine or any third-application that communicates with it, allowing a party to submit one or more rogue policies to the system. This last type of attack cannot be prevented by the blockchain, and thus relies on good security practices on edge applications and devices.

What are the privacy implications of having a transparent ACL system? As the data stored in the blockchain is available to all nodes that belong to it, some privacy concerns can arise. However, since the approach consists of a consortium blockchain, all the nodes are, *a priori*, well-known and trustworthy, mitigating the privacy concerns of a public blockchain.

IV. Implementation Details

During the implementation of the *proof-of-concept*, some decisions were made; we describe these details here with the intent of helping the reader to re-implement a similar *proof-of-concept* or production-ready system. The whole system, from the transactions logic to the writing to the blockchain, was implemented using `Javascript` due to the simplicity of the language and availability of libraries. Such libraries, as for example, the built-in `crypto` module provides us with a mechanism to digitally sign the blocks payload using public-key cryptography (using RSA-SHA256) and to calculate the hashes of each block using SHA256 algorithm. From our perspective and due to the huge disparity among data types, three main classes were created: `Document`, `Entity` and `Transaction`.

The information of the State Machines mentioned in section III-A is given by the `Transactions` data present in each moment and not by the states of the machine itself. Thus, the current state of the system given by the process of *snapshot*. A *snapshot* is like a picture taken from the blockchain, consisting of applying all previous transactions until now in order to get the state of all operations. As example, for evaluate a access request requires the execution of all transactions about that specific 3^{rd} `Party`, `Record` and associated `Data Keepers`.

The implemented blockchain uses *proof-of-work* that bases itself on a *brute-force* mechanism of hash calculation. This mechanism works as the *nonce* is iteratively incremented until the resulting SHA256 hash matches the *a priori* defined number of leading zeroes — this is similar to the Bitcoin system and establishes the "effort".

However, we can easily tweak the "effort" to better suit our use-case. Although we implemented the *proof-of-work* mechanism, due to its simplicity, there are alternatives as *Proof of Elapsed Time* (PoET) or *proof-of-stake* mechanisms, that are not as resource consuming as the implemented one, which can be better to validate the transactions in the context of a consortium blockchain applied to eHealth scenarios.

One of the details that required some attention was the writing of *blocks* to the blockchain. As previously mentioned, each transaction has a unique identifier (`ID`) to allow the identification of the same transaction over its different states. This helps to prevent different states of the same transaction from being written to the same block.

A local data storage system is used to maintain the blockchain data, in order to reduce the possibility of needing to query the P2P network for the whole blockchain in case of a fault in the `Node` and to keep information about the last snapshot (as way of improving performance as the chain grows in size).

V. Sanity Checks

For the purpose of testing and validating the approach, a preliminary sample running scenario was assembled with the deployment of the proof-of-concept along a simple distributed architecture. Tests covered network disruption scenarios between `Blockchain Nodes` and node failures and recoveries.

The *facilities* were simulated, each running a `Blockchain Node` in different but connected machines. For the purposes of simulating such system architecture, each micro-service was mounted as a Docker container and the network configured using the Docker network manager.

A set of different access control transactions were submitted to different nodes, and verifications were pursued by observing and checking the snapshots with the expected outcome.

Each facility is submitted to the following seven-phased test suit (sorted by processing order):

1. `IsAlive`: Tests if the *Blockchain Node* is up and running;
2. `AddRules`: Add a set of access policies to the batch stored locally (in cache) and assures its correct addition;

3. `CommitRules`: Pushes the local batch of policies to the `Blockchain Node`, which processes it (`mine`) and publish to the network. The local cache batch is cleared after this;

4. `Snapshot`: Makes a snapshot of the current state of the blockchain, assuring that all the policies correspond with the *a priori* defined when they were commited;

5. `AddRule`: Add a single and overriding access policies to the local batch;

6. `CommitRules`: Pushes the local batch of access control policies to the `Blockchain Node`, which processes it (*mine*) and publish to the network. The local batch is cleared after that;

7. `Snapshot`: Makes another snapshot of the current state of the blockchain, assuring that the override over the previous access policy was correctly processed.

This seven-phased test suit is run on all nodes, testing the correct function of the blockchain when submitting *quasi*-simultaneous and different transactions to it.
Additionally, faults were manually *injected* in random nodes, by the means of stopping and/or restarting the Docker containers, verifying the resilience of the blockchain to networks and/or machine faults.
Two simulated environments were tested, one consisting of three facilities and another one consisting of ten facilities. In each one of these simulated distributed architectures, the seven-phased test suit was run.
Each execution of the test suit (one *per* facility) submits a total of two blocks to the blockchain. Since the blockchain have a *genesis-block* when it is initialized, the total of blocks in the blockchain is given by the formula: $totalOfBlocks = 1 + 2 * numberOfNodes$.
So, after running each test scenario, we get a blockchain with a total of:

- 7 blocks in the 3 facilities test scenario;
- 21 blocks in the 10 facilities test scenario.

Showing that the blocks are being correctly added to the blockchain, validating the correct store of the access control policies, checking the sanity of our approach.

VI. Conclusions

In this paper, we presented an approach to solving the problem of managing access control in the eHealth ecosystem. Access Control is a special complex task in eHealth since resources and data are distributed among different facilities and institutions. Further, this is even more problematic because in some cases, eHealth resources are not owned or managed by a single entity or individual. As a way of overcoming this complexity, we propose an approach that leverages the use of blockchain for store transactional information about eHealth records and access control policies.
For purposes of supporting the plausibility of the scheme proposal, a *proof-of-concept* was designed and implemented. This *proof-of-concept* allowed us to make some, even if preliminary, tests and validations over the sanity of the approach from a functional and applicational perspective.
Overall, we determine that the approach is viable, giving diverse advantages when comparing to the in-place systems. These advantages includes, but are not limited to, the integrity, transparency, and authenticity of the access control policies in the system, being, this information distributed and synchronized by all the institutions and organizations that make part of the consortium.
Further, research needs to be pursued in order to make such an approach ready to be used in real scenarios. In this context, further testing and validation are needed to assess the scalability proprieties of such system. This includes testing large-scale scenarios with different node dynamics, i.e., adding, removing and invalidating nodes on-the-fly. Also, tests dealing with malicious attacks by third-parties or cases when one or more nodes of the blockchain are compromised should be pursued.

Acknowledgments

This work was supported by Project NanoSTIMA: Macro-to-Nano Human Sensing: Towards Integrated Multimodal Health Monitoring and Analytics/NORTE-01-0145-FEDER-000016" financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF).

References

[1] P. Lukowicz, T. Kirstein, and G. Troster, "Wearable systems for health care applications," *Methods of Information in Medicine-Methodik der Information in der Medizin*, vol. 43, no. 3, pp. 232–238, 2004.

[2] K. Patrick, W. G. Griswold, F. Raab, and S. S. Intille, "Health and the mobile phone," *American journal of preventive medicine*, vol. 35, no. 2, p. 177, 2008.

[3] M. N. K. Boulos, A. Rocha, A. Martins, M. E. Vicente, A. Bolz, R. Feld, I. Tchoudovski, M. Braecklein, J. Nelson, G. Ó. Laighin, *et al.*, "Caalyx: a new generation of location-based services in healthcare," *International journal of health geographics*, vol. 6, no. 1, p. 9, 2007.

[4] IDC, "The digital universe: Driving data growth in healthcare," report, EMC Corporation and International Data Corporation, 2014.

[5] W. Raghupathi and V. Raghupathi, "Big data analytics in healthcare: promise and potential," *Health Information Science and Systems*, vol. 2, no. 1, p. 3, 2014.

[6] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, "The internet of things for health care: a comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.

[7] P. witek and A. Rucinski, "Iot as a service system for ehealth," in *2013 IEEE 15th International Conference on e-Health Networking, Applications and Services (Healthcom 2013)*, pp. 81–84, Oct 2013.

[8] L. Tan and N. Wang, "Future internet: The internet of things," *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, vol. 5, pp. V5–376–V5–380, Aug 2010.

[9] T. Sahama, L. Simpson, and B. Lane, "Security and privacy in ehealth: Is it possible?," in *2013 IEEE 15th International Conference on e-Health Networking, Applications and Services (Healthcom 2013)*, pp. 249–253, IEEE, 2013.

[10] P. C. Tang, J. S. Ash, D. W. Bates, J. M. Overhage, and D. Z. Sands, "Personal health records: definitions, benefits, and strategies for overcoming barriers to adoption," *Journal of the American Medical Informatics Association*, vol. 13, no. 2, pp. 121–126, 2006.

[11] NAHIT, "Report to the office of the national coordinator for health information technology on defining key health information technology terms," tech. rep., Office of the National Coordinator for Health Information Technology, April 2008.

[12] V. C. Hu, D. Ferraiolo, and D. R. Kuhn, *Assessment of access control systems*. US Department of Commerce, National Institute of Standards and Technology, 2006.

[13] W. O. Nijeweme-d'Hollosy, L. van Velsen, M. Huygens, and H. Hermens, "Requirements for and barriers towards interoperable ehealth technology in primary care," *IEEE Internet Computing*, vol. 19, pp. 10–19, July 2015.

[14] J. P. Dias, H. S. Ferreira, and n. Martins, "A blockchain-based scheme for access control in e-health scenarios," *Proceedings of the 13th International Conference on Information Assurance and Security (IAS)*, 2018.

[15] Deloitte., "Bitcoin, Blockchain & distributed ledgers: Caught between promise and reality," tech. rep., Centre for the Edge, Australia, 2015.

[16] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," p. 9, 2008.

[17] A. M. Antonopoulos, *Mastering Bitcoin: unlocking digital cryptocurrencies*. " O'Reilly Media, Inc.", 2014.

[18] F. Ethereum, "Ethereum: A next-generation smart contract and decentralized application platform," 2014.

[19] C. Cachin and M. Vukolic, "Blockchain consensus protocols in the wild," *CoRR*, vol. abs/1707.01873, 2017.

[20] S. Underwood, "Blockchain beyond bitcoin," *Commun. ACM*, vol. 59, pp. 15–17, Oct. 2016.

[21] R. Lai and D. L. K. Chuen, "Blockchain–from public to private," in *Handbook of Blockchain, Digital Finance, and Inclusion, Volume 2*, pp. 145–177, Elsevier, 2018.

[22] M. M. Hossain, M. Fotouhi, and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the internet of things," in *2015 IEEE World Congress on Services*, pp. 21–28, IEEE, 2015.

[23] R. E. Scott, P. Jennett, and M. Yeo, "Access and authorisation in a Glocal e-Health Policy context," *International Journal of Medical Informatics*, vol. 73, no. 3, pp. 259–266, 2004.

[24] A. Dogac, T. Namli, A. Okcan, G. Laleci, Y. Kabak, and M. Eichelberg, "Key issues of technical interoperability solutions in ehealth and the ride project," *Software R&D Center, Dept. of Computer Eng., Middle East Technical University, Ankara*, vol. 6531, 2007.

[25] M. Li, S. Yu, K. Ren, and W. Lou, "Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings," *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, pp. 89–106, 2010.

[26] V. C. Hu, D. Ferraiolo, R. Kuhn, A. R. Friedman, A. J. Lang, M. M. Cogdell, *et al.*, "Guide to attribute based access control (abac) definition and considerations (draft)," *NIST special publication*, vol. 800, no. 162, 2013.

[27] S. Godik and T. Moses, "Oasis extensible access control markup language (xacml)," *OASIS Committee Secification cs-xacml-specification-1.0*, 2002.

[28] J. Bogaerts, M. Decat, B. Lagaisse, and W. Joosen, "Entity-based access control: Supporting more expressive access control policies," in *Proceedings of the 31st Annual Computer Security Applications Conference*, ACSAC 2015, (New York, NY, USA), pp. 291–300, ACM, 2015.

[29] M. Mettler, "Blockchain technology in healthcare: The revolution starts here," in *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pp. 1–3, Sep. 2016.

[30] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K. R. Choo, "Blockchain: A panacea for healthcare cloud-based data security and privacy?," *IEEE Cloud Computing*, vol. 5, pp. 31–37, Jan 2018.

[31] P. Zhang, D. C. Schmidt, J. White, and G. Lenz, "Chapter One - Blockchain Technology Use Cases in Healthcare," in *Blockchain Technology: Platforms, Tools and Use Cases* (P. Raj and G. C. Deka, eds.), vol. 111 of *Advances in Computers*, pp. 1–41, Elsevier, 2018.

[32] D. Di Francesco Maesa, P. Mori, and L. Ricci, "Blockchain based access control," *Distributed Applications and Interoperable Systems: 17th IFIP WG 6.1 International Conference*, pp. 206–220, 2017.

[33] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare Data Gateways: Found Healthcare Intelligence on Blockchain with Novel Privacy Risk Control," *Journal of Medical Systems*, vol. 40, no. 10, 2016.

[34] A. Chepurnoy and D. Meshkov, "On space-scarce economy in blockchain systems," vol. 2017, p. 644, 2017.

[35] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE international congress on big data (BigData congress)*, pp. 557–564, IEEE, 2017.

Author Biographies

João Pedro Dias He has an MSc in Informatics and Computing Engineering by the Faculty of Engineering, University of Porto. He is a Ph.D. student of the Doctoral Program in Informatics Engineering by the same university since 2017. He is an Invited Assistant Lecturer at FEUP since 2017 and has co-supervised 4 MSc dissertations. Has participated as Researcher in 2 projects at LIACC and INESC TEC (Porto, Portugal). Works in the area of Software Engineering, with a special interest in Design Patterns, Internet-of-Things, and Security.

Hugo Sereno Ferreira He has a Ph.D. in Informatics by the Universities of Porto, Aveiro, and Minho in Portugal. Former Postdoctoral Research at INESC TEC, where hes now a Research Associate. Assistant Professor at the Faculty of Engineering, University of Porto (FEUP), since 2008 of more than 20 different curricular units. His main research areas (+50 published works) include Large-Scale Software Systems, Design and Architectural Patterns, Machine Learning and Distributed Ledger Technologies (Blockchain), where he supervised more than 60 students in these topics.

Ângelo Martins He has a Ph.D. in Computer Engineering - Information Systems by Porto University. Senior researcher and co-coordinator of the Computer Graphics and Information Systems unit of INESC TEC. 30 years experience in software development and 25 years experience in teaching software development at Porto Polytechnic School of Engineering. For 10 years he was program manager of the Informatics Engineering Bologna 1st cycle program (Bach). Involved in several projects in the e-health area: Caalyx (FP6), eCaalyx (AAL JP), AAL4ALL (national).

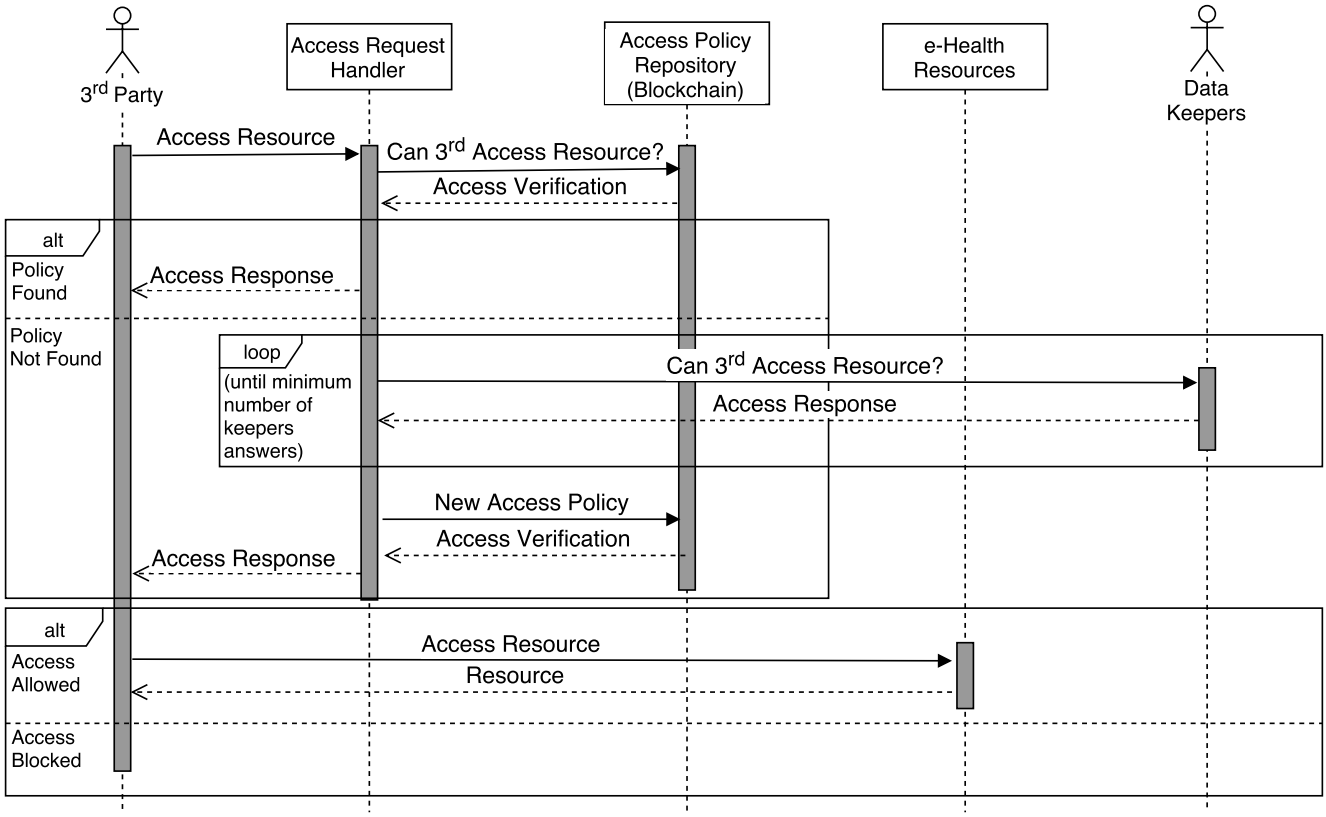

Figure. 12: Sequential view on how some `3rd Party` can access or request access to an eHealth resource, detailing the communication between the inner modules of the architecture. It is also visible the process of creation of new access control policies.